- _LineCoord (int x1, int y1, int x2, int y2),
- _Line (point p1, point p2);

```
typedef struct {
        int x, y; } point;
```

- _PolyLineCoord (int vertexcount, int *xArray, int *xArray);
- _PolyLine (int vertexcount, point *vertices);

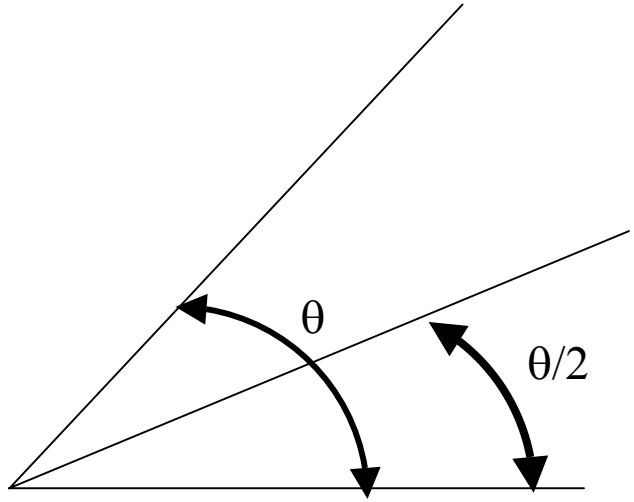
- _MarkerCoord (intx, int y);
- _Marker (point pt);
- _PolyMarkerCoord (int vertexcount, int *xArray, int *xArray);
- _PolyMarker (int vertexcount, point *vertices);

- _Polygon (int vertexcount, point *vertices);
- _RectangleCoord (int leftX, int bottomY, int rightX, int topY);
- _RectanglePt (point bottomleft, point topright);
- _Rectangle (rectangle rect);

- **_defpoint (int x, int y);**
- **_defrectangle(int leftX, int bottomY, int rightX, int topY);**

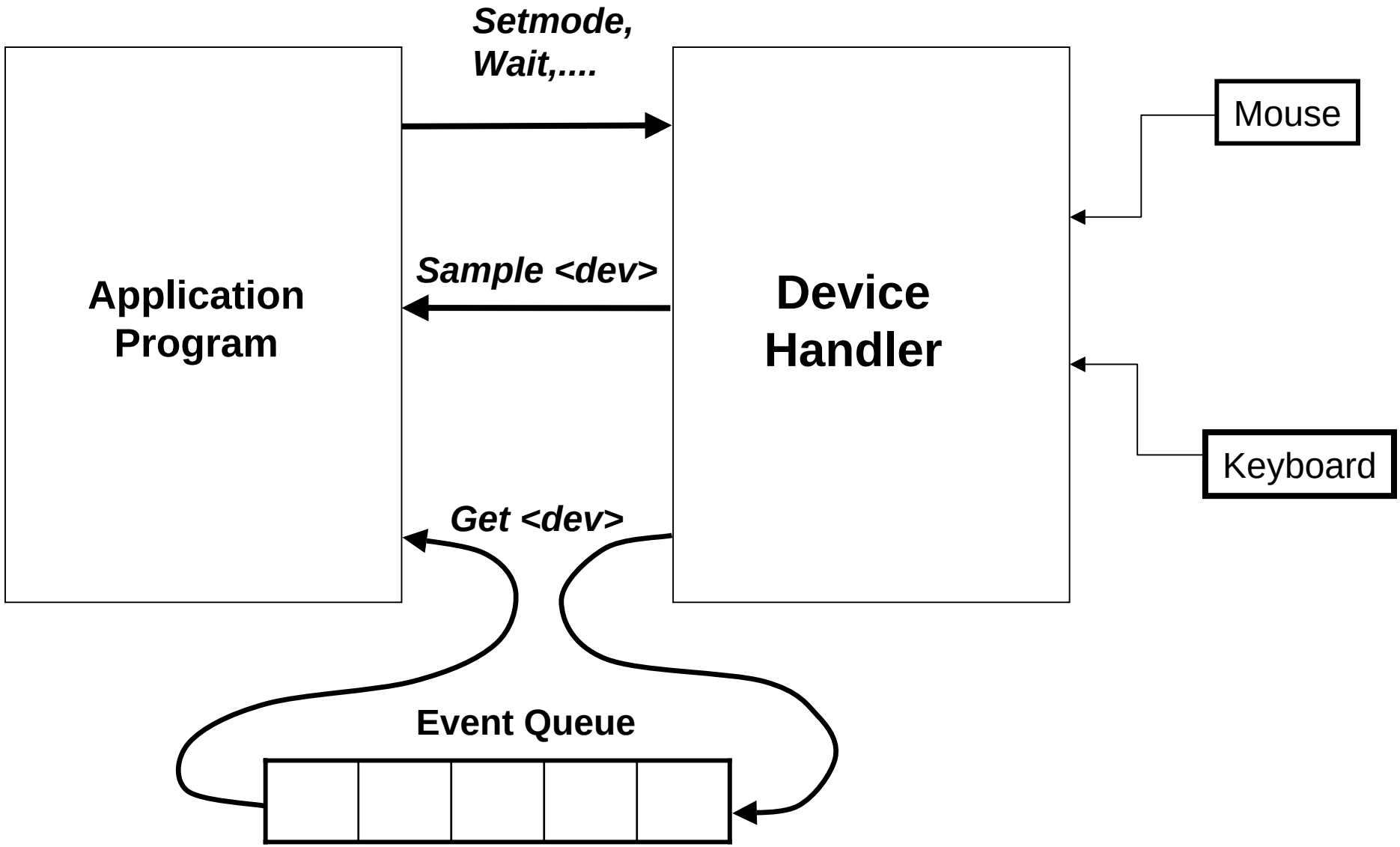
- **_ellipseArc (rectangle extentrect, double startangle, double endangle);**

What about an angle bisector ?



- **_setLinestyle (CONT/DASHED/DOTTED/.....);**
- **_setLineWidth (int width);**
- **_setMarkerSize (int markersiyze);**
- **_setMarkerStyle (CIRCLE/SQUARE/STAR/.....);**
- **_setColor (int colorindex);**

Sampling vs. Event-handling using the event Queue



Event-driven interaction scheme

```
Initialization calls;
activate <interactive dev> in event mode;

while (no request from user – Quit)
        wait for the event to be triggered from any device

        switch(<dev. causing interrupt>)

                case <dev1>: collect data, respond #1;
                case <dev2>: collect data, respond #2;
                case <dev3>: collect data, respond #3;
                .
                .
        endswitch
endwhile
```

Frame-buffers

Monochrome : 1 bit per pixel (bitmap)

Full color: 24 bits/pixel - 8 for each or r,g,b

Others: if a LUT is not used, can have as many colors or shades of grey as specified by the number of bits/pixel

8 bits => 256 colors (normally 3,3,2) or shades of grey.

Color LUTs (palettes)

Each entry in the frame buffer is an index into the LUT.

- if n bits/pixel => 2n entries in the LUT

LUT entry then determines the color sent to the screen.

If each LUT entry is p bits, then can display 2p possible colors (example p=24 => 16 million colors in the palette)

Can only display 2n colors simultaneously.

Example (typical) :

- **frame-buffer: 8 bits/pixel**
- **LUT: 24 bits/entry**
- **Therefore, can display 256 colors at any one time out of a possible 16 million**

Advantages of using LUT?

- cheaper than full color (3 bytes*1280*1024 = almost 4MB of memory)

- allows more displayable colors than without one.

- Can do color table animation